Insight in Cyber Safety when Remotely Operating SCADA Systems of Dutch Critical Infrastructure Objects

Tina Tami
University of Amsterdam
Amsterdam, Netherlands
tina.tami@os3.nl

Supervisor
Cedric Both
DataDigest
The Hague, Netherlands

***Abstract*—SCADA systems of Dutch critical infrastructure objects such as bridges and tunnels are vulnerable to malware attacks and exploits. Such attacks could have catastrophic consequences such as the flooding of an area. Numerous anomaly detection techniques are available for detecting such exploits, though few to none of them use event-based data. After considering multiple state-of-the-art methods for anomaly detection, a Markov chain model is proposed. When experimenting using event logging from the Velsertunnel, illegal sequences of events were easily detected, whereas legal sequences which are possibly anomalous are more difficult to detect.**

I. Introduction

Supervisory Control and Data Acquisition (SCADA) systems are used for collecting, forwarding, processing and visualizing measurement and control signals from different machines in large industrial systems [1], including Dutch Critical Infrastructure (CI) objects. Such objects include bridges, tunnels and water facilities like locks. In the past, it was a common misconception that the SCADA networks were electronically isolated from all other networks [2]. People invested much of their time and effort in increasing its physical security, expecting that attackers would not be able to access the systems from the outside. However, the increasing inter connectivity of SCADA networks has made them vulnerable to various network security problems and has therefore raised concerns regarding their cyber safety [3]. Because of this increase in inter connectivity, the industrial control systems of Dutch CI objects are likely to suffer from cyber attacks, malware and exploits [4]. Such exploits could affect the operation of these objects negatively, having large-scale consequences. Imagine the damage caused of an intruder tampering with the water lock, causing the area to flood.

Ideally, the SCADA systems should be able to detect anomalies where human control is impractical, in order to provide decision support for the managing parties involved in controlling the objects. This would be of great help in detecting abnormal behaviour of the objects with regard to preserving their cyber safety. Since research has shown that the security of the Dutch CI objects is poorly organized, the Ministry of Infrastructure and Water Management (Rijkswaterstaat) guaranteed the House of Representatives to map these security flaws and make improvements [5].

The data streams of these Dutch CI objects are shown in figure 1. The objects are remotely controlled by the control party from central traffic centers, where they observe these objects on monitors and remotely operate them when needed. This could be in order to move the object in a certain way because of its functionality, such as opening and closing the bridge for a boat to pass by, or in order to repair disruptions and malfunctions. In addition to the control party, there is a maintenance party. Their role is to physically visit the object when needed to carry out either planned or unplanned maintenance. There are numerous data sources, mainly being sensor data, logging data and network data. All data is sent back and forth to centralized systems from where the control party remotely operates the objects. The maintenance party can access the same data locally from within the object, or remotely via an external VPN connection through the centralized network. Both these parties use this data to control, maintain and observe the object. Each of these objects has SCADA, Programmable Logic Controller (PLC), servers, applications, CCTV and audio-equipment.

For this research, logging data will be used that also includes some sensor data. Network data will be left out of scope because this data is scanned and monitored by the Security Operations Center (SOC). This is a facility where an information security team monitors and analyzes the network security of Rijkswaterstaat, in order to respond to incidents. To do this, they use Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS). We will look at the data streams that involve the communication between the SCADA systems, maintenance party and control party. The scope of this research is to use existing data and not focus on researching new techniques for gathering more data. The goal is to detect anomalies in behaviour and events regarding human actions and the state of the object.

A. Research questions

The main question for this research is defined as follows: *How can anomalies be detected in event data from SCADA*

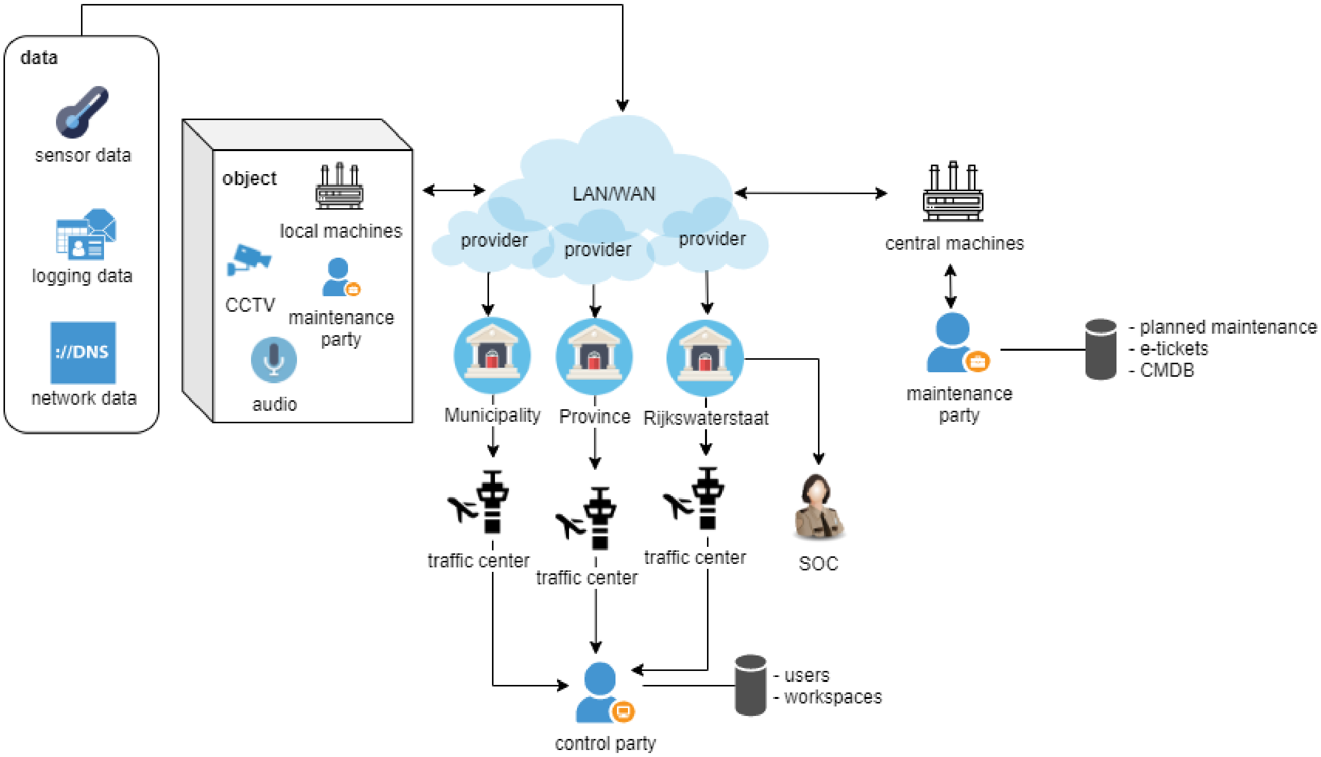

Figure 1. High-level view of the data streams, involved systems and involved parties.

and other involved systems, in order to provide security alerting and decision support rules?

This question is further divided into the following sub-questions:

1) What data is relevant and necessary in order to decide if an action is considered anomalous?
2) What techniques can be used to detect anomalies and which one would be best suited for event-based data?

B. Structure

In the upcoming section, section II, we will look at related work done on the topic of anomaly detection in SCADA systems. In section III we discuss the requirements for this research as well as a few considered anomaly detection techniques. A breakdown will be provided of each technique and its applicability to this research. The dataset and its preparation will be explained in section IV, along with the chosen approach. Section V shows our obtained results which will be further discussed in section VI. The conclusion that can be drawn from this research will be stated in section VII. Finally, we discuss points for future research on this topic in section VIII.

II. Related Work

Research on the state of security in SCADA systems has shown that services such as anti-virus and firewalls are turned off by users in order to increase the accuracy of the SCADA system, since they may slow down the flow of data or drop packets [6]. Turning these services off exposes the operating systems to viruses and malware, which are threats to the systems. [6] showed that it is important to provide operators with the correct expertise, as issues have arisen in the past where proper training of people working with SCADA systems was missing. Incorrectly configuring the systems and misreporting of information can leave the system vulnerable to attacks. Poor user administration can also be the cause of exploits, since not removing employees’ access on the termination of a contract grants them access to the entire system. This can lead to disastrous results, as in 2000 an ex-employee managed to get access to the waste management system and thereby caused millions of litres of raw sewage to spill out into local parks and rivers [7]. Nowadays, a large part of the risk lies in (sub)contractors not modifying default passwords. These passwords can be found in online documentation, granting an intruder effortless access to penetrate the SCADA systems [8]. Regarding Dutch CI objects, incidents could occur where intruders gain access to the system this way and perform unwanted actions. Therefore, we would like to be able to detect these anomalous actions.

Much research has been done on the topic of anomaly detection in SCADA systems using artificial intelligence [9] [10] [11] [12]. A commonly used technique for detecting anomalies is using neural networks. In [13], these networks were used to establish prediction models of the condition parameters used in wind turbines. These parameters were dependent on different environmental conditions such as ambient temperature and wind speed. Their results showed that the proposed method was more effective in anomaly identification in wind turbines than traditional methods [13]. The use of neural networks will be considered in this research as well. Even though anomaly detection in SCADA networks has been extensively researched, methods for using event-based data have rarely been proposed. Thus, further examination is needed in order to select the most promising method for this research.

III. Methodology

This section will provide an overview of requirements to successfully detect anomalies, as well as a few considered anomaly detection techniques and machine learning methods. Each technique is briefly explained and examined on its applicability for this research.

A. Requirements

The definition of an anomaly is dependant of its context. In some cases it can be said that the occurrence of a certain action is anomalous. In event-based data, however, an individual action cannot be anomalous by itself, as each logged event is capable of occurring in an object under regular circumstances. Therefore, other factors need to be taken into account in order to detect whether an action is anomalous or not. These other factors might be time and location, or the person who is responsible for the event taking place. A legal event can become illegal when performed by an actor who is not qualified or expected to carry out the action. Imagine a bridge opening at an unusual time, regarding its regular use. If we know that the actor of this possibly anomalous action is a member of the maintenance party, the corresponding agenda could be consulted in order to see whether there is scheduled maintenance. Based on this information, an appropriate response can be suggested to provide decision support for the managing party.

The chosen approach and method for detecting these anomalies has to satisfy certain criteria. Firstly, it has to be able to manage textual input, as event based data is not numerical. Even though the logging data used for this research partly consists of sensor data, it is not displayed in a numeric manner. Secondly, it must take into account the previous event(s), as we established that in our case, an event cannot be anomalous by itself. Finally, the proposed method has to be applicable using real-time data. Since the goal is to detect anomalies and provide decision support, there is no interest in detecting anomalies in retrospect.

B. Anomaly Detection using Machine Learning

Anomaly detection is an important data analysis task which is useful for identifying all sorts of abnormal behaviour and events from a given dataset by using machine learning techniques [14]. There are numerous different anomaly detection algorithms, each of them being more effective within a specific context. We will discuss a few of the considered models and explain its applicability for this research.

1) Clustering-based algorithms: Clustering-based algorithms are categorized as unsupervised machine learning algorithms which do not require pre-labeled data to extract rules for grouping similar data instances [15]. There are many well known algorithms such as *k-means clustering* [16] and *Gaussian mixture modeling* [17], which work well with numerical data. Such algorithms work under the assumption that we can create clusters of only normal data, and new data that do not fit well with existing clusters of normal data are considered anomalous [14]. Since these algorithms are solely used for numerical data, we do not consider them good candidates for anomaly detection in event-based data sets. Therefore, we will not use these algorithms for our experiment.

2) Support Vector Machine using TF-IDF: The basic principle of the Support Vector Machine (SVM) is to derive a hyperplane that maximizes the separating margin between the positive and negative classes [18]. SVM on its own requires numerical data, though it can be used with textual data by using Term Frequency-Inverse Document Frequency (TF-IDF). The combination of these techniques can be used to reflect the importance of a word in a specific corpus, having many applications such as movie recommendation based on movie plots. Although this method does work well with textual data, such findings do not support our desire for detecting anomalies in events.

3) Neural Networks: Long Short-Term Memory (LSTM) neural networks are a sub-type of the more general recurrent neural networks (RNN) [19]. LSTM networks have been demonstrated to be particularly useful for learning sequences containing longer term patterns of unknown length, due to their ability to maintain long term memory [20]. In order to use neural networks with textual data, events could be split in separate categories and encoded using *one-hot encoding* [21], where each entry will be converted into an index. The drawback using this method is that the neural network performs poorly if a categorical variable takes on a large number of values [21]. Considering the data set that will be used in this research, which has a vast amount of unique values and will be further illustrated in section IV, LSTM neural networks are dismissed as a recommended method.

4) Markov Chain Modeling: A Markov chain is a stochastic model describing a sequence of possible events which holds the Markov Property. This property states that given the entire history of the subject, the present state depends only on the most recent past state [22]. In other words, the probability of transitioning to any particular state is dependent solely on the current state. This can be visualized in a diagram as shown in figure 2. Markov chain models can and have been used for detecting anomalies in a sequence of observations [23], and are not restricted to using numerical data. Taking the latter into account along with the fact that the model describes sequential observations, it is expected that this could be a promising method to use in this research. This hypothesis will be further examined in section IV.

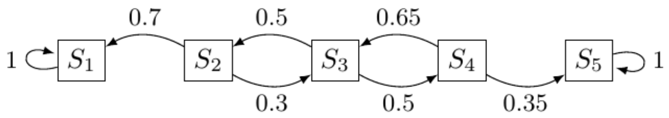

Figure 2. Example of a Markov chain diagram showing five states and their probabilities of moving to another state [24].

IV. Approach

This section covers the preparation of the used data set, in addition to a detailed description of our approach to detect anomalies using a Markov chain model.

A. Data preparation

As mentioned earlier, we intent to describe a method to detect anomalies in event-based data. The data set that is used for experimenting is retrieved from logging data captured from the Velsertunnel, a cross-river connection under the North Sea Canal carrying over 60,000 vehicles per day. The time range of the event data is 12 hours, which contains over 40 thousand entries. Each entry consists of 15 fields which are displayed in table I. Unfortunately, some of these fields are rarely or never defined, or all set to the same value. To illustrate, *user* is only set in 8% of the data, with its value always being 1. Therefore, such fields are omitted, as well as fields that do not provide relevant information, such as repetitions of multiple fields combined. The remaining, relevant fields that are included are *timestamp*, *subsystem*, *text* and *location*. Initially, the data set embodies 24 unique subsystem entries and 2191 unique text entries. In order to create a smaller subset for testing, the focus will lie on the entries with the subsystem being *SUS*, *CON* and *HD*. The SUS subssystem, which stands for speed under-run system, contains sensors that detect slowly moving or stationary traffic. In addition to this, HD subsystems consist of height detectors that get triggered by the passing of a vehicle exceeding the height limit. The control events executed by the control and maintenance party are displayed as the CON subsystem. These subsystems together show a straightforward sequence of events, resulting in a saturated data set. In this case, *saturated* is used to express that every possible event that can take place in the object is included, which does not apply to most other subsystems in the initial data set. This leaves us with 23 unique *text* entries. Regarding this smaller dataset, a rather simplified version of what happens in general is displayed in figure 3. Each row can be seen as a single sequence, where *lff* stands for *logical function fulfiller* and *OL* stands for *operating location*. It is important to note that a sequence can occur within another sequence as there are multiple height detectors and SUS sections which can be triggered successively. However, the logical function fulfiller should always be triggered after the local operating location has performed an action, without the interference of another event.

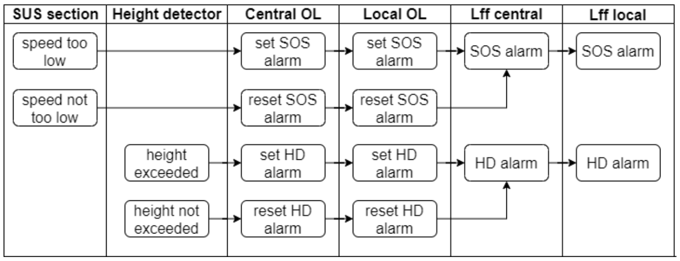

Figure 3. A simplified illustration of the sequential events taking place inside the processed data set. Each row can be seen as a single sequence.

B. Set-up

Now that the processed data set has been established, we can define the application of Markov chain modeling. We consider a sequence of events $\{X_1, X_2, \ldots, X_n\}$ with s distinct states. A Markov chain can then be characterized by an $s \times s$ matrix containing the transition probabilities P_{ij}, called a *transition probability matrix*:

$$\text{P} = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1s} \\ P_{21} & P_{22} & \cdots & P_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ P_{s1} & P_{s2} & \cdots & P_{ss} \end{bmatrix},$$

where

$$P_{ij} = P(X_{t+1} = j | X_t = i) \tag{1}$$

and

$$P_{ij} = n_{ij}/n_i \tag{2}$$

[25]. P_{ij} is derived from the number of observed consecutive transitions from state *i* to state *j*, divided by the number of times that state *i* is observed. Its value can range between 0 and 1. By mathematically modeling these sequences, we can characterize normal behaviour and therefore detect unusual behaviour. We can split the event logging data set into a training set and a test set. The training set will be used to create the transition probability matrix, which we will refer to as TPM, and the test set will be used as new event logging. Since our processed data contains 23 possible events, TPM is a 23 × 23 matrix that describes the probabilities of moving to another state, given a current state. When a new event is logged, we can estimate the probability of that event happening given its previous state by looking for those consecutive events in our TPM as shown in figure 4. Keep in mind that all possible states should and will be mentioned in TPM as there is a finite number of events that can take place. Therefore, a probability value will always be returned and this does not mean it is an anomaly by definition. In order to detect anomalies, a threshold should be set to only report a sequence of events that have a probability of happening lower than that threshold. This means that setting the threshold to 1 will return all events in pairs, ranked from lowest probability to highest probability. The pairs returned with the lowest probability are not necessarily

Table I
ONE ENTRY FROM THE EVENT LOGGING DATA SET WITH THE CORRESPONDING FIELDS.

timestamp	**event value**	**control module**	**subsystem**	**object type**	**function element**	**text**
17-1-2020 08:03:14:100	2	Li	VLV	sfDeelverlichting	Variabelen	Verlichting gesloten deel: HandStand

tag name	**type**	**location**	**camera**	**note**	**arguments**	**user**	**workspace**
Li_bfVerlichtingVb_sfDeelverlichtingG.Variabelen.HandStand	Variabelen	HB-Li					

anomalies, but it might be worth to further investigate them in order to see if some illegal activity is happening. Because the event logging data set contains regular behaviour, no events will be reported back from this test set if the threshold were to be 0.

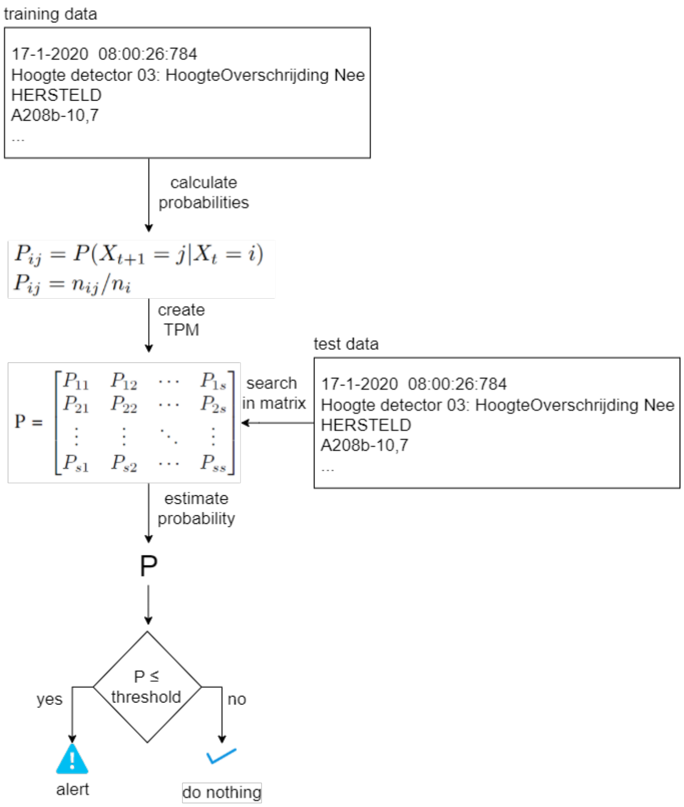
training data
17-1-2020 08:00:26:784
Hoogte detector 03: HoogteOverschrijding Nee
HERSTELD
A208b-10,7
...
calculate probabilities
$P_{ij} = P(X_{t+1} = j|X_t = i)$
$P_{ij} = n_{ij}/n_i$
create TPM
test data
$\mathbf{P} = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1s} \\ P_{21} & P_{22} & \cdots & P_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ P_{s1} & P_{s2} & \cdots & P_{ss} \end{bmatrix}$
search in matrix
17-1-2020 08:00:26:784
Hoogte detector 03: HoogteOverschrijding Nee
HERSTELD
A208b-10,7
...
estimate probability
P
P ≤ threshold
yes
no
alert
do nothing

Figure 4. The selected approach using Markov chains. Training data is used to create the TPM. When a new event is logged, we can estimate the probability of that event happening given its previous state by looking for those consecutive events in our TPM. If this probability is lower than the selected threshold, the sequence of events will be returned.

In order to test this method, various anomalies will be inserted in the test set manually, so we know what should be returned when looking for anomalies. Such a manually inserted anomaly could be a number of things. As mentioned before, the logical function fulfiller should always be triggered after the local operating location has performed an action. If otherwise, these events should be returned, even with the threshold set to 0. Variations based on these events not happening consecutively are definite anomalies, since they are simply not allowed to happen and do not happen under regular circumstances. In addition to these definite anomalies, there are some sequences that are not anomalies per se. When looking at the SUS events, an SUS alarm should only be reset by the central operating location after the speed issue has been resolved by the SUS section. If this sequence is not being executed in this order, then it might be the case that (1) someone performed an unjustified action, (2) an intruder is tampering with the system, or (3) there is a valid reason. Either way, it should be reported as potentially anomalous for further investigation. Note that the difficulty level of the anomaly detection increases here: the resetting of the alarm does not have to take place immediately after resolving the speed issue, as there are multiple SUS sections that could be triggered independently. Therefore, these sequences of events are not expected to be returned when the threshold is set to 0. We will evaluate multiple different inserted anomalies and their position in the returned list of plausible anomalies. This means that if a sequence of events occurs in the test set that has a probability of 0 in the TPM, it will be returned on position 1.

V. RESULTS

Multiple variations of anomalous sequences were manually inserted in the test set. The threshold is set to 1 so a list is returned of all sequential entries, ranked from lowest probability to highest probability according to our previously established TPM. The output of this anomalous test set is compared to the unaltered test set. The result of one of these tests, where five anomalies were manually inserted, is shown in table III. The anomalies that are inserted are shown in table II. As expected, sequences of events that do not take place in the training set but do take place in the altered test set are detected perfectly. Unfortunately, few things can be said about the inserted sequences with a probability larger than 0. We can see that the five inserted anomalies are detected on position 1, 2, 3, 5 and 6. In this specific case, one could say that there is only one false positive in the anomalous test set (number 4 in table III), if we consider that all five anomalies have been returned on the sixth position. However, this is training set-dependant as these inserted anomalous sequences apparently do have a relatively low probability of happening compared to other sequences of events. Moreover, most of the inserted anomalous sequences apparently do not appear in the unaltered test set, which explains why they are not returned on the left-hand side of the table.

VI. DISCUSSION

Note that even though the method applied to gather results uses a data set as input, it is feasible to use a single event

Table II
FIVE MANUALLY INSERTED ANOMALOUS SEQUENCES. A DESCRIPTION IS PROVIDED TO EXPLAIN WHY THE SEQUENCE IS ANOMALOUS.

Insertion	Description
17-1-2020 17:42:03:781 SUS sectie 204: Snelheid te laag A22-11,4-HRR 17-1-2020 17:42:04:000 Bedienlocatie Centraal: ResetAlarmContact SUS TN	Central OL resets SUS alarm after SUS section detects a too low speed.
17-1-2020 17:53:39:788 SUS sectie 218: Snelheid te laag A22-12,4-HRL 17-1-2020 17:53:39:854 lfvBediening Centraal: SUS	Lff logs a change in SUS alarming even though no OL performed this action beforehand.
17-1-2020 18:07:59:998 Bedienlocatie Lokaal: SetAlarmContact SUS TN 17-1-2020 18:08:18:861 Snelheid te laag HERSTELD A22-12,9-HRR	Local OL sets an alarm without the lff logging this change afterwards.
17-1-2020 18:27:56:932 lfvBediening Centraal: SUS 17-1-2020 19:32:38:961 Bedienlocatie Centraal: SetAlarmContact DefHoogteDetectie TN	Central OL sets an HD alarm without a height detector being triggered beforehand.
17-1-2020 19:32:38:961 lfvBediening Lokaal: DefHoogteDetectie 17-1-2020 19:32:58:438 Bedienlocatie Centraal: ResetAlarmContact DefHoogteDetectie TN	Central OL resets HD alarm without the height detector reporting the height is no longer exceeded.

entry as would be the case in real-life application. Thus, it is not necessary to gather a data set of a specific size in order for this method to work. Because of its estimating nature, the method proposed is difficult to evaluate as events will always be returned when the threshold is set to 1. Naturally, the threshold has to be lower than 1, but choosing this number is a complex yet crucial task. If the chosen threshold is too high, valuable time will be wasted investigating non-anomalies, whereas setting the threshold too low will result in ignorance of illegal behaviour. The ideal way to measure its performance would be to hypothesise a well-considered threshold and test its accuracy in determining anomalies by applying it using real-time event logging. The drawback, however, is that testing with (Dutch) CI objects is costly and leaves no room for experiments possibly resulting in disruptions. Moreover, careful investigation is required when selecting a training set. As the key feature for detecting possible anomalies is using the transition probability matrix, it is of great importance to use a saturated data set to compute the matrix. In addition, the selected data set cannot contain anomalous behaviour, since this would cause the matrix to contain probabilities that are not valid under regular circumstances.

VII. Conclusion

As this research utilises event-based data, it is necessary to have knowledge of each event with its corresponding time, location and actor. Moreover, a general overview is required which maps all agents with their responsibilities, qualifications and agenda. Possessing this information eases the task of defining and detecting anomalies. After considering numerous well-known anomaly detection techniques, a Markov chain model is proposed in order to detect anomalies in event based data from Dutch CI objects. The conducted experiment using event logging from the Velsertunnel showed that illegal sequences of events were easily detected, whereas legal sequences which are possibly anomalous are more difficult to detect due to the threshold variable. Even though it can be said that the detection of such anomalies is more complicated, it is difficult to measure and therefore state its actual performance. Furthermore, crucial data is missing in order to provide meaningful decision support.

VIII. Future Work

The proposed method of using Markov chains could be improved by designing a theory to estimate the best-performing threshold. Doing this will limit the managing party to follow up on actual potential anomalies. Moreover, it will allow performance testing in order to measure its working and possibly compare it to other techniques. Additionally, more data has to be retrieved and consequently implemented in order to provide decision support, such as the actor of an event and their qualifications and agenda.

Table III

OUTPUT OF THE TEST SET EXCLUDING ANOMALIES NEXT TO THE OUTPUT OF THE TEST SET INCLUDING ANOMALIES. THE TRAINING SET WAS USED TO CREATE THE TPM WHICH ESTIMATES THE SHOWN PROBABILITIES.

Position	**Test set excluding anomalies**	**Test set including anomalies**
1	Current state: IfvBediening Centraal: SUS Next state: HoogteOverschrijding Ja Probability of this happening: 0.0215053763 Time of anomaly: 17-1-2020 18:27:56:932 Location of anomaly: -	Current state: Snelheid te laag Next state: IfvBediening Centraal: SUS Probability of this happening: 0.0 Time of anomaly: 17-1-2020 17:53:39:788 Location of anomaly: A22-12,4-HRL
2	Current state: IfvBediening Centraal: SUS Next state: HoogteOverschrijding Ja Probability of this happening: 0.0215054 Time of anomaly: 17-1-2020 10:45:40:456 Location of anomaly: -	Current state: Bedienlocatie Lokaal: SetAlarmContact SUS Next state: Snelheid te laag HERSTELD Probability of this happening: 0.0 Time of anomaly: 17-1-2020 18:07:59:998 Location of anomaly: TN
3	Current state: IfvBediening Lokaal: SUS Next state: Bedienlocatie Centraal: ResetAlarmContact SUS Probability of this happening: 0.0698925 Time of anomaly: 17-1-2020 17:53:39:854 Location of anomaly: nan	Current state: IfvBediening Centraal: SUS Next state: Bedienlocatie Centraal: SetAlarmContact DefHoogteDetectie Probability of this happening: 0.0107526882 Time of anomaly: 17-1-2020 18:27:56:932 Location of anomaly: -
4	Current state: IfvBediening Lokaal: SUS Next state: Bedienlocatie Centraal: ResetAlarmContact SUS Probability of this happening: 0.0698925 Time of anomaly: 17-1-2020 18:12:17:34 Location of anomaly: -	Current state: IfvBediening Centraal: SUS Next state: HoogteOverschrijding Ja Probability of this happening: 0.0215054 Time of anomaly: 17-1-2020 10:45:40:456 Location of anomaly: -
5	Current state: IfvBediening Lokaal: SUS Next state: Snelheid te laag Probability of this happening: 0.0913978494623656 Time of anomaly: 17-1-2020 17:42:01:899 Location of anomaly: nan	Current state: Snelheid te laag Next state: Bedienlocatie Centraal: ResetAlarmContact SUS Probability of this happening: 0.0260870 Time of anomaly: 17-1-2020 17:42:03:781 Location of anomaly: A22-11,4-HRR
6	Current state: Snelheid te laag HERSTELD Next state: Snelheid te laag HERSTELD Probability of this happening: 0.11403508771929824 Time of anomaly: 17-1-2020 17:42:21:832 Location of anomaly: A22-12,9-HRL	Current state: IfvBediening Lokaal: DefHoogteDetectie Next state: Bedienlocatie Centraal: ResetAlarmContact DefHoogteDetectie Probability of this happening: 0.0454545 Time of anomaly: 17-1-2020 19:32:38:961 Location of anomaly: -

REFERENCES

[1] *Supervisory control and data acquisition*. 2018. URL: https://nl.wikipedia.org/wiki/Supervisory_control_and_data_acquisition (visited on 02/02/2020).

[2] R. Carlson. "Sandia SCADA Program – High Surety SCADA LDRD Final Report". In: (2002). DOI: 10.2172/800787.

[3] V.M. Igure, S.A. Laughter, and R.D. Williams. "Security issues in SCADA networks". In: *Computers Security* 25.7 (2006), pp. 498–506. DOI: 10.1016/j.cose.2006.03.001.

[4] N. Castellon and E. Frinking. *Securing Critical Infrastructures in the Netherlands*. 2015. URL: https://www.thehaguesecuritydelta.com/media/com_hsd/report/53/document/Securing-Critical-Infrastructures-in-the-Netherlands.pdf (visited on 01/09/2019).

[5] S. van Gils. "Kamer eist preventieve cyberaanval op sluizen en stormkeringen". In: *Financieel Dagblad* (2019). URL: https://fd.nl/economie-politiek/1304087/tweede-kamer-eist-preventieve-cyberaanval-op-sluizen-en-stormkeringen.

[6] A. Nicholson, S. Webber, S. Dyer, T. Patel and H. Janicke. "SCADA security in the light of Cyber-Warfare". In: *Computers Security* 31.4 (2006), pp. 418–436. DOI: 10.1016/j.cose.2012.02.009.

[7] T. Smith. "Hacker jailed for revenge sewage attacks". In: *The Register* (2001). URL: https://www.theregister.co.uk/2001/10/31/hacker_jailed_for_revenge_sewage/.

[8] A. Belqruch and A. Maach. "Proceedings of the 2nd International Conference on Networking, Information Systems Security". In: Association for Computing Machinery, 2019. DOI: 10.1145/3320326.3320328.

[9] D. Gamez J. Bigham and N. Lu. "Safeguarding SCADA Systems with Anomaly Detection". In: 2003, pp. 171–182. DOI: 10.1007/978-3-540-45215-7_14.

[10] A. Zaher et al. "Online wind turbine fault detection through automated SCADA data analysis". In: *Wind Energy* 12.6 (2009), pp. 574–593. DOI: 10.1002/we.319.

[11] D. Yang, A. Usynin, and J. Hines. "Anomaly-Based Intrusion Detection for SCADA Systems". In: (2008).

[12] P. Jarventausta et al. "AI-based methods in practical fault location of medium voltage distribution feeders". In: 1996, pp. 164–169. DOI: 10.1109/ISAP.1996.501062.

[13] P. Sun et al. "A generalized model for wind turbine anomaly identification based on SCADA data". In: *Applied Energy* 168 (2016), pp. 550–567. DOI: 10.1016/j.apenergy.2016.01.133.

[14] M. Ahmed, A.N. Mahmood, and J. Hu. "A survey of network anomaly detection techniques". In: *Journal of Network and Computer Applications* 60 (2016), pp. 19–31. DOI: 10.1016/j.jnca.2015.11.016.

[15] A. K. Jain, M. N. Murty, and P. J. Flynn. “Data Clustering: A Review”. In: *ACM Comput. Surv.* 31.3 (1999), pp. 264–323. DOI: 10.1145/331499.331504.

[16] N. Vlassis A. Likas and J.J. Verbeek. “The global k-means clustering algorithm”. In: *Pattern Recognition* 36.2 (2003), pp. 451–461. DOI: 10.1016/S0031-3203(02)00060-2.

[17] Douglas Reynolds. “Gaussian Mixture Models”. In: *Encyclopedia of Biometrics*. 2009, pp. 659–663. DOI: 10.1007/978-0-387-73003-5_196.

[18] Eleazar Eskin et al. “A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data”. In: *Applications of Data Mining in Computer Security* 6 (2002). DOI: 10.1007/978-1-4615-0953-0_4.

[19] B. Larzalere. *LSTM Autoencoder for Anomaly Detection*. URL: https://towardsdatascience.com/lstm-autoencoder-for-anomaly-detection-e1f4f2ee7ccf (visited on 01/28/2020).

[20] Pankaj Malhotra et al. “Long Short Term Memory Networks for Anomaly Detection in Time Series”. In: *ESANN*. 2015.

[21] *Using categorical data with one-hot encoding*. URL: https://www.kaggle.com/dansbecker/using-categorical-data-with-one-hot-encoding (visited on 01/28/2020).

[22] Bruce A. Craiga and Peter P. Sendib. “Estimation of the transition matrix of a discrete-timeMarkov chain”. In: *Health Econ.* 11 (2002), pp. 33–42. DOI: 10.1002/hec.654.

[23] A. R. Breurker et al. *Real-time anomaly detection in critical Rabobank processes*. URL: https://studiegids.tue.nl/opleidingen/bachelor-college/majors/technische-bedrijfskunde/bachelor-end-project-bep/ (visited on 01/14/2020).

[24] URL: https://tex.stackexchange.com/questions/268797/illustrate-transitions-between-states-in-markov-chain (visited on 01/09/2019).

[25] I. Teodorescu. *Maximum Likelihood Estimation for Markov Chains*. Tech. rep. arXiv:0905.4131. May 2009. URL: https://cds.cern.ch/record/1179509.